

**AN INVESTIGATION OF THE USE OF TEMPORAL DECOMPOSITION
IN SPACE MISSION SCHEDULING**

Submitted By

**Stanley F. Bullington, Ph.D., P.E.
Venkat Narayanan**

**Department of Industrial Engineering
Mississippi State University
P.O. Drawer U
Mississippi State, MS 39762**

**In Fulfillment of the Requirements for
NASA Grant NAG8-1012
NASA George C. Marshall Space Flight Center**

July, 1994

ABSTRACT

This research involves an examination of techniques for solving scheduling problems in long-duration space missions. The mission timeline is broken up into several time segments, which are then scheduled incrementally. Three methods are presented for identifying the activities that are to be attempted within these segments. The first method is a mathematical model, which is presented primarily to illustrate the structure of the temporal decomposition problem. Since the mathematical model is bound to be computationally prohibitive for realistic problems, two heuristic assignment procedures are also presented. The first heuristic method is based on dispatching rules for activity selection, and the second heuristic assigns performances of a model evenly over timeline segments. These heuristics are tested using a sample Space Station mission and a Spacelab mission. The results are compared with those obtained by scheduling the missions without any problem decomposition. The applicability of this approach to large-scale mission scheduling problems is also discussed.

TABLE OF CONTENTS

	Page
ABSTRACT	ii
LIST OF TABLES	v
CHAPTER I. INTRODUCTION	1
CHAPTER II. LITERATURE REVIEW	4
Space Mission Scheduling Problems	4
Characteristics	4
Constraints and Requirements	7
Temporal Constraints	7
Sequencing Constraints	8
Relational Constraints	9
Resource Requirements	10
Crew Requirements	10
Solution Approaches	11
The Scheduling Process of ESP	11
Selection Methods	13
Loading Algorithm	14
Temporal Decomposition	15
CHAPTER III. MATHEMATICAL MODEL FOR TEMPORAL DECOMPOSITION	21
Pre-processing	21
Mathematical Model for Assignment	24
CHAPTER IV. HEURISTIC ASSIGNMENT PROCEDURE FOR TEMPORAL DECOMPOSITION	29
Assignment Heuristic	29
Activity Selection Rules	33
CHAPTER V. EXPERIMENTAL RESULTS	36
Introduction	36
Space Station Mission Results	38
Spacelab Mission Results	40
General Comments	42

	Page
CHAPTER VI. SUGGESTIONS FOR FUTURE RESEARCH	45
General Comments	45
Possible Use of Artificial Neural Networks	48
CHAPTER VII. CONCLUSIONS	53
REFERENCES	55

LIST OF TABLES

	Page
TABLE 1. Results For Space Station Mission	39
TABLE 2. Results For Spacelab Mission	41

CHAPTER I

INTRODUCTION

Scheduling is the assignment of limited resources over time to perform a set of tasks (Baker, 1974). Scheduling problems arise naturally in many systems and are of immense practical significance. Even many "simple" scheduling problems, however, have been shown to belong to the hardest class of mathematical problems, known as the "NP-hard" class (Ullman, 1976 and Garey and Johnson, 1979).

Space mission scheduling problems typically involve a multitude of activities. Activities require multiple resources and are restricted by several types of constraints which should be satisfied simultaneously. Adding to this difficulty is the inherent (stochastic) nature of the domain to defy predictions. In long-term planning problems, many constraints cannot be predicted accurately. Space Station scheduling problems, in particular, present even greater complexity due to the length of the missions involved. A natural way to handle this difficulty is to schedule in an incremental fashion.

Several compelling reasons exist for scheduling long missions in an incremental (segmented) fashion. The size of the planning problems may make them computationally intractable. The scheduling difficulty would be compounded due to the large numbers of activities and constraints involved in a long mission. For long missions, the likelihood of rescheduling due to unforeseen developments increases

significantly. Such rescheduling is accomplished more efficiently when planning has been done in a segmented manner. Furthermore, missions such as those of the Space Station can be divided into natural increments defined by the arrivals of shuttles to the station.

The capability to decompose a mission timeline into segments ("macrowindows") is available in the Experiment Scheduling Program (ESP) used at NASA's Marshall Space Flight Center. Since there is generally no need for performing temporal decomposition in Spacelab mission planning, the macrowindows capability is not typically used for this purpose. However, in anticipated long-duration missions, segmented scheduling will be necessary. Therefore, a study of temporal decomposition in space mission scheduling problems is indicated.

The aim of this research is to investigate means for effectively performing segmented scheduling. Intelligent means of allocating activities to different mission increments have been studied. The candidate solution techniques developed have been evaluated on some simple, but realistic problems. The result of the experimental work will hopefully provide useful information regarding the benefits of segmented scheduling, and regarding promising means of carrying out such scheduling.

Chapter II presents some general characteristics and constraints of space mission scheduling problems, including solution approaches used, followed by a review of literature relating to segmented scheduling. In Chapter III, a mathematical model for assigning activities to mission increments is presented, primarily in order to

illustrate the structure of the temporal decomposition problem. Since the mathematical model is computationally intractable for realistic problems, Chapter IV deals with heuristic assignment procedures. The first heuristic method is a loading algorithm based on dispatching rules for activity selection. The second heuristic assigns model performances evenly over time segments. In Chapter V, experimental results are given for sample Spacelab and Space Station missions, followed by a discussion of the performance of the loading algorithms. The applicability of the proposed approaches to large-scale mission scheduling problems is also discussed. In Chapter VI, some suggestions are given for future research on temporal decomposition of space missions. Conclusions are presented in Chapter VII.

CHAPTER II

LITERATURE REVIEW

Space Mission Scheduling Problems

The scientific and operational environments are different for various space missions. Consequently, the scheduling objectives and solution techniques used tend to be unique. However, there are some similarities among the different types of space mission scheduling problems, and in the type of constraints that restrict these problems (time windows, for instance). Bullington and Jaap (1992) provide a comparison of mission scheduling and production scheduling in terms of the scheduling environment, objectives and solution methods.

Characteristics

Space mission scheduling problems are static problems in that the entire set of tasks, along with the constraints, are generally known in advance. The tasks to be scheduled are called "models" or activities. A model consists of several "steps" or operations which are to be done in a required order. A model may have to be replicated several times; these replications are called "performances." No two steps (or performances) of the same model should be active simultaneously.

Mission scheduling problems are subject to three broad types of constraints, namely resource constraints, precedence constraints, and temporal constraints, which

place a restriction on when an activity can be executed. An obvious way of enhancing schedule quality is to increase the number of activities that are performed in parallel, while simultaneously satisfying these constraints.

When more demands are placed on the resources than can be allocated, the resulting problems are referred to as "over-subscribed" scheduling problems. In mission scheduling, both the number of models and the number of performances of certain models are over-subscribed. The latter enables prioritization by providing a high selection probability for such models. However, the number of model performances actually scheduled must not exceed the specified maximum.

In production scheduling, deadlines are generally considered relaxable, even as they are constraints placed on a job's due date. Minimizing job tardiness is a common objective. In mission scheduling, the preference (soft) constraints are often the only relaxable constraints (Smith and Pathak, 1992). However, these should be satisfied as much as possible since schedule quality depends on the degree of fulfillment of both hard and soft constraints.

The optimization criteria can influence the scheduling complexity significantly. For instance, in single-machine scheduling, optimizing for the flow-time criterion is polynomial, while optimizing for the tardiness criterion is NP-hard. Mission scheduling problems are multi-criteria optimization problems; the scheduling goals are often conflicting, and may change with time. Two common objectives are to maximize scientific return and resource utilization. Scientific return can be

maximized by increasing the number of models scheduled, and by scheduling as many high priority (critical) models as possible (Gaspin, 1989).

When a solution that meets all constraints and objectives does not exist, it is better to achieve one that provides the best overall compromise (Smith and Pathak, 1992). To accomplish this, the scheduling process must have low computational requirements, thus enabling the generation of many trial schedules. The schedule has to be periodically refined to overcome the effects of various dynamic factors such as "targets of opportunity" and unexpected resource breakdowns. Schedule repair and rescheduling techniques are employed to maintain and, if possible, improve schedule quality, in such circumstances.

The complexity of mission scheduling problems, thus, is due to the number and types of constraints, optimization criteria and stochasticity. Furthermore, the scheduling difficulty in a space station is compounded by its long mission duration, which typically leads to a significant increase in the number of activities to be scheduled. A space station's mission duration is expected to be more than a decade (Goldin, 1993). For such missions, the computational overheads placed on the scheduler would be extremely high if scheduling is done in a non-decomposed fashion. This may, thus, force the use of some type of temporal decomposition, or segmented scheduling.

Constraints and Requirements

The various constraints and requirements specified by models are outlined in this section (Mission Planning Division, 1993 and Stacy and Jaap, 1988).

Temporal Constraints

Performance time windows define the time frames within which the performances of a model can be executed. Each window specifies a start time, an end time, and the maximum number of performances that can be scheduled within the time frame. A model may indicate a preference to be scheduled at certain intervals within its time window. Time windows may overlap or be intermittent. Models (steps) cannot be performed outside these windows. Opportunity windows arise due to a celestial or terrestrial target and/or attitude of the spacecraft. Macrowindows are user-specified time windows.

Performance duration is the required operation time of one performance of a model. Step duration is the operation time of a step, and defines the required period of usage for any resource, crew, target and/or attitude specified by the step.

Performance separation time is the time delay between adjacent performances of a model. Likewise, step delay is the time lag between a step and the previous step of a model. The delays, step durations and time windows are specified as a minimum and a maximum which must not be violated. The actual performance duration of a model

(the sum of the actual step durations and step delays) must not exceed the stated maximum duration.

Scenarios are alternate orderings of the steps of a model. The production scheduling equivalent is the existence of alternate routes to produce certain jobs. A scenario consists of a list of steps, their order of execution, a priority rating, and a selection strategy. Certain steps may be repeated in a scenario while certain others may not be included. A selection strategy defines the conditions under which the scenario can be selected.

A target represents a condition or opportunity required for scheduling a model step. Target requirements are usually environmental in nature (the visibility of a celestial or terrestrial target, for instance). A step can be scheduled only when all specified targets are available. An attitude represents a requirement of a step on the spatial orientation (inclination) of the spacecraft. A step may also specify that it not be scheduled when a specific target or orbital opportunity is available.

Sequencing Constraints

Sequencing constraints specify the models that are to precede and/or succeed a particular model. A step or model may be confined to start/end within a specified time relative to another step or model (sequencing delays). If more than one performance is requested for the required model, the sequenced model can be executed with any of the performances of the required model. Multiple performances

can be scheduled following a performance of the required model. Thus, a model with predecessors can start as soon as one performance of each required model is complete; its earliest start time is the maximum of the completion time of the first performance of the predecessors. Note, however, that activities in a mission do not form a network as in project scheduling problems.

Relational Constraints

A model step may specify that it be performed concurrently with a step of some other model. Concurrency may be mandatory, necessary or desired. In the case of mandatory concurrency, neither step (model) can be executed without the other. That is, both steps should be scheduled together; otherwise, neither can be scheduled. Selection of one model for scheduling forces selection of the other. For necessary concurrency, if both cannot be scheduled together, the concurrent step (the one requiring concurrency) is not scheduled while the required step is unaffected. If concurrency is desired, a step may prefer to be executed with another model step. The concurrent step should be scheduled with the required step, if possible. However, if this is not possible, the concurrent model can still be scheduled.

Scheduling of a concurrent model (step) is deferred if it is selected before the required model. Also, irrespective of the type of concurrence, it is generally preferred that the two steps start together. A step or a model may also specify that it should not be scheduled concurrently with some other step or model.

Resource Requirements

A step may use three types of resources, namely consumables, non-depletables and equipment, and can be scheduled only when all specified resources are available. Consumables are those resources whose availability is permanently decreased when they are utilized (for example, energy, photographic film, etc.). Non-depletables are those whose availability is decreased only for the interval of usage (for example, power, crew time, etc.). When a step is scheduled, the availability of a non-depletable resource is decreased by the amount of usage, which is replenished once the step is complete. A step may also use some piece of equipment which is not available to other models until the step is completed (a TV camera, for instance). Resource carry-through enables resource usage to continue through the step delay to the next step. Models must be scheduled such that the total resource usage, at any instant, does not exceed the total availability.

Crew Requirements

Crew members may be required to perform, or (periodically) monitor model steps. A step may require specific crew members or may enable a choice between several persons. The latter flexibility in crew ordering can be used to balance crew utilization to a certain extent. Crew balancing is performed on a per-performance basis rather than on a per-step basis; if possible, the same members are utilized to perform several related steps ("crew lock-in").

Solution Approaches

The complexity of space mission scheduling problems requires the use of diverse techniques to address different problems. Thus, within the same problem domain, it is quite common to use a method that limits the search space over all possible schedules, while another method is used to resolve resource assignment conflicts (Thalman et al., 1991). Solution techniques designed to find optimal solutions are generally unsuitable in mission scheduling problems due to their high computational requirements. For instance, Sheskin (1988) developed a zero-one integer programming formulation for scheduling experiments in the Space Station, and solved an example problem consisting of two activities and ten time periods. The computational requirements of such a model is bound to be prohibitive for realistic instances. Since optimal schedules form a small subset of all possible schedules, it is generally advisable to search for near-optimal schedules.

Many systems are available to NASA for scheduling space missions. Of these, ESP is one of the most powerful and popular ones. It is a generic system, and has been used for scheduling numerous Spacelab missions. It is also the host scheduler for this research. The scheduling process used in ESP is outlined below.

The Scheduling Process of ESP

ESP selects an activity using dispatching rules, and constructs the schedule one model performance at a time. The scheduling of a model depends on the selection

order and the satisfaction of the model's constraints. In ESP, a multitude of activities vie for limited resources. The selection order has a profound effect on schedule quality since resources are assigned on a "first-come, first-served" basis. Once a model is selected, the time periods available for the model are checked to determine the time at which it can be scheduled with respect to the constraints. One performance of the selected model is then placed on the timeline, and the resource and crew availabilities are suitably adjusted. The process is repeated until all performances of all models have been attempted (Jaap and Davis, 1988, 1989). The quality of a timeline depends on the extent to which it accomplishes stated mission goals. In general, a schedule that satisfies many different criteria is preferred over another which fares well for only a few performance measures. ESP judges the quality of a schedule using the schedule grade function which incorporates five different criteria (Stacy and Jaap, 1988).

$$\text{Schedule Grade} = \frac{(w_1P + w_2A + w_3C + w_4T + w_5S)}{100},$$

where

$$P = \frac{\text{Number of Performances Scheduled}}{\text{Number of Performances Requested}},$$

$$A = \frac{\text{Number of Activities Scheduled}}{\text{Number of Activities Requested}},$$

$$C = \frac{\text{Crew Time Utilized}}{\text{Crew Time Requested}},$$

$$T = \frac{\text{Activity Operation Time}}{\text{Minimum Activity Time Requested}},$$

$$S = \frac{\text{Science Value Scheduled}}{\text{Science Value Requested}},$$

and w_1 , w_2 , w_3 , w_4 and w_5 are user-specified weights for the various criteria. Science value gives the scientific value of a step relative to the mission's expected value.

Selection Methods

Activities are generally grouped in terms of their discipline or experiment nature. Various selection methods can be used within these groups such as (Mission Planning Division, 1993): (1) fixed order selection, in which the user pre-specifies the exact selection order (static), (2) random order selection, where each model performance has an equal probability of selection, (3) maximize grade selection, which selects a model that will cause the greatest increase in the schedule grade, (4) most-constrained selection, in which the most time-constrained models in a fixed-order group are attempted first, and (5) manual selection, in which the user dynamically dictates the next model to be attempted.

Random selection is quite popular among system users. Grade-maximization is a gradient (dynamic) selection method wherein the selection of a model depends on

the current schedule quality and the grade of models available for scheduling. As in production scheduling and project scheduling, no rule has been found to be robust under a variety of conditions (see Blackstone et al., 1982, Boctor, 1990, and Maxwell, 1987). Several trial schedules can be generated by varying the selection order of model groupings, and by changing the selection rule; the schedule that yields the "best" value for the schedule grade function is chosen.

Loading Algorithm

ESP uses forward chaining, and depth-first search with backtracking to place one performance of the selected model on the timeline without violating any constraint (Stacy and Jaap, 1988). The scenario to be employed is determined by the selection method, based on the scenario strategy and priority rating.

The loading algorithm first determines the specific times at which the constraints stated by a model step are satisfied. The constraints are checked in a depth-first fashion ("nested windows") until a low-level search window where the step can be scheduled is found. If any constraint is violated, checking proceeds to the next search window. If all windows are exhausted, the model step is failed. A model performance is scheduled only when all steps have been successfully loaded. Once scheduled, models cannot be shifted or unscheduled.

Front loading of models is preferred due to the requirement that models start as soon as possible, unless specified otherwise. This serves to schedule as much

science as practical early in the mission which is a particularly important objective due to the possibility of a premature mission termination, or other unforeseen event (Bullington and Jaap, 1992). Also, front loading results in the building of a semi-active schedule wherein no job can be performed earlier without altering the sequence; Baker (1974) has shown that at least one optimal schedule is semi-active. Maximizing step (activity) duration is preferred over maximizing the number of performances since the former does not involve any increase in time lost due to setup and stowage.

Temporal Decomposition

Problems associated with large systems are usually solved by decomposing the system into connected or disconnected sub-systems. The complexity of a large problem can be vastly reduced by decomposing it. The resource allocation problem in a production system can, for instance, be split into independent dispatching problems in the individual workcenters (Chryssolouris et al., 1991). Other types of such non-temporal decomposition have been used frequently in scheduling problems (see Yamamoto, 1977 and Chu et al., 1992).

Temporal decomposition involves breaking up the mission timeline into non-overlapping segments, and identifying the models that are to be attempted within these segments. Models should be assigned to segments such that they have adequate opportunity for being scheduled, and such that their constraints and requirements can

be satisfied while scheduling. Since several time choices exist for a model, the decomposition process should place an activity at a good temporal position so as to enable the scheduling of as many models as possible. Once such a decomposition has been obtained, the various segments are scheduled in an incremental fashion. Sadowski and Jacobson (1978) have shown that an optimal allocation of tasks in each segment does not yield an optimal overall solution.

Little work has been done regarding segmented scheduling of space missions, with the possible exception of the work by Machuca and Sadowski (1981), and the SPIKE system developed for Hubble Space Telescope (HST) scheduling (Miller et al., 1988 and Johnston and Adorf, 1992).

Machuca and Sadowski developed a scheduling system for NASA's satellite communications network in which the problem was treated as a generic resource-constrained scheduling problem with time windows and over-subscribed resources. Since there were no precedence constraints in this problem, the timeline was split into segments. Two strategies, namely the BASIC and the MAX procedure, were tested.

The BASIC procedure uses a sequencing approach in which jobs are attempted to be scheduled in the order given by ranking rules. The MAX procedure utilizes ranking rules and partial enumeration techniques to find the 'best' sequence. The zero-one integer programming formulation, for multi-project scheduling, developed by Pritsker et al. (1969), was adapted. Various job rankings and segment durations were

tested based on the merits of the sequences they produced; the MAX procedure was found to be the superior one.

A key distinction between communications scheduling and space mission scheduling is the lack of sequencing and relational constraints in the former. Also, a partial enumeration approach is likely to be impractical for mission scheduling due to the large number of activities.

In the SPIKE scheduling system (Johnston and Adorf, 1992), observations that are to be performed contiguously are "clustered" together so as to limit the number of entities, thus reducing the problem size. A "cluster" is the smallest assignable entity, and is assigned to start during the time interval of a segment. Multiple clusters may be assigned to a segment, but a cluster can be committed to only one segment. Activities within a cluster are not required to end within a segment.

"Suitability functions" were used to represent the level of satisfaction of the constraints of an activity at a segment. Thus, these determine the desirability of starting an activity at a segment by providing evidence for/against scheduling decisions. The satisfaction of hard constraints was measured using constraint propagation techniques, and the hard constraints were combined with the soft requirements using evidential reasoning techniques. A cluster can be assigned to a segment only if all activities within the cluster have non-zero suitabilities at that segment. Detailed scheduling is done every week, during which the clusters and

constraints are fully expanded. The schedule quality is measured by the "summed suitability function."

Johnston and Adorf (1992) presented a zero-one mathematical formulation for HST scheduling. This model was transformed to a static, timetable-type neural network along with the apt biases and connection strengths. A neuron represents the allocation of a particular cluster (row) to a particular segment (column). The linear equality and inequality constraints of the mathematical formulation were modeled using "guard neurons" which destroy the symmetry of the model. Asymmetric feedback networks, as duly noted by the authors, have not yet been proven to be asymptotically stable. The network can, however, be used without any training, and attempts to find the maximal independent set of assignments. By controlling the feedback dynamics of the network, both predictive and reactive scheduling can be done. Several algorithms were developed using the "suitability functions" framework. Of these, the neural encoding was found to be the fastest, and permits the most exhaustive exploration of the solution space.

Miller and Johnston (1991) presented several methods for performing segmented scheduling. A "procedural scheduler", which commits clusters to segments based on greedy algorithms, is described. These algorithms were found to schedule clusters at times of low suitability, thus creating poor schedules. A modified Hopfield neural network was used, in which clusters were assigned to segments upon satisfying the constraints. The insight obtained from the neural network model was

used to develop a constraint satisfaction formulation which was found to perform better than the network formulation. Miller et al. (1988) remark that the "summed suitability function" does not provide adequate discrimination between schedules since it neglects the effects of diminishing resources within segments while scheduling the clusters.

In SPIKE, as noted above, temporal decomposition is done in two stages: (1) clustering the activities, and (2) assigning the clusters to segments. Alternatively, activities can be assigned directly (i.e., in a single stage) to the segments. This may lead to a balanced assignment, and thereby a better schedule, owing to the added flexibility in assigning individual activities, rather than clusters of activities. Besides the contrast in the degree of approximation, the SPIKE scheduling system and the assignment algorithm given here (see Chapter III) are distinct in terms of the solution methods used and the scheduling architecture employed.

HST is merely an observatory in space and the models to be scheduled are basically telescopic experiments; this enables observations on similar targets to be grouped together. In a space station, however, observation is simply one of the many tasks, and there is likely to be a great degree of variety in the tasks. The number and types of constraints that affect models in a space station is bound to be much higher than those experienced in HST. The amount of parallelism expected in a space station, for example, is much more than that encountered in the telescope.

Segmented scheduling can be performed, in ESP, by specifying the macrowindows of a model as segments in which it is to be attempted; provision is available to define the number of performances that are to be attempted within these macrowindows. It has been determined by NASA that a space station mission timeline should be divided into week-long segments. Given the size of the space station scheduling problem, the decomposition process should have low computational requirements, thus permitting the evaluation of many different task orderings; a possible slight degradation in schedule quality may be permissible in exchange for computational savings (Bullington and Jaap, 1992).

In a segmented scheduling scenario, there is likely to be a significant reduction in the scheduling time of ESP since: (1) the number of models that compete for selection, in a segment, is limited due to temporal decomposition; this may lead to a reduction in the selection time, and (2) the search required for loading a model performance, in a segmented mission, is limited to be within the time length of a segment; whereas, in a non-decomposed mission, ESP may have to search the entire timeline before being able to load a model performance.

CHAPTER III

MATHEMATICAL MODEL FOR TEMPORAL DECOMPOSITION

Pre-processing

Consider a space mission with $i = 1, \dots, M$ activities, where each activity, i , has $q = 1, \dots, S_i$ steps, and has to be replicated N_i times (i.e., multiple performances). Each step, q , of model i has a minimum and maximum step duration, $tmin_{iq}$ and $tmax_{iq}$, respectively, and a minimum and maximum step delay, $dmin_{iq}$ and $dmax_{iq}$, respectively, with the previous step of the model (when $q = 1$, the delays are zero). Since the steps must be performed contiguously, model performances can be thought of as being rendered in a single step; step-level modeling is likely to make the decomposition tedious. The minimum and maximum performance durations of model i are, respectively,

$$Pmin_i = \sum_{q=1}^{S_i} (tmin_{iq} + dmin_{iq})$$

$$Pmax_i = \sum_{q=1}^{S_i} (tmax_{iq} + dmax_{iq}).$$

If several scenarios exist for a model, the one with the highest requirements or maximum weight is chosen. The actual performance duration for model i , P_i , is

between P_{\min_i} and P_{\max_i} . The minimum and maximum performance delays for model i are given by D_{\min_i} and D_{\max_i} , respectively.

A mission has a total of $g = 1, \dots, G$ crew members, and $b = 1, \dots, B$ resources (renewable and non-renewable). Based on the minimum and maximum step durations, each step, q , of model i requires NC_{iq} crew members, each for $\{c_{\min_{iq}}, c_{\max_{iq}}\}$ unit hours. Therefore, model i requires NC_i crew members, where $NC_i = \max_q \{NC_{iq}, 1 \leq q \leq S_i\}$. Then, each performance of model i requires between c_{\min_q} and c_{\max_q} unit hours of each of the NC_i crew members, where

$$c_{\min_i} = \sum_{q=1}^{S_i} c_{\min_{iq}}$$

$$c_{\max_i} = \sum_{q=1}^{S_i} c_{\max_{iq}}$$

Similarly, each step, q , of model i requires between $r_{\min_{iqb}}$ and $r_{\max_{iqb}}$ unit hours of resource type b . The minimum and maximum resource usages of a model performance of model i are, respectively,

$$r_{\min_{ib}} = \sum_{q=1}^{S_i} r_{\min_{iqb}}$$

$$r_{\max_{ib}} = \sum_{q=1}^{S_i} r_{\max_{iqb}}, \quad \text{for all } i \text{ and } b.$$

Each model i specifies a set of crew members $Cset_i$, where $n(Cset_i) \geq NC_i$ (i.e., crew ordering is generally flexible). Crew usage also includes crew lock-in, and resource usage also includes resource carry-through. The total resource and crew availability through the length of the mission is R_b unit hours, for all b , and C_g unit hours, for all g (if all members are equally available, $C_g = C$, for all g), respectively. Only the most constraining resources b need to be considered. The target, attitude, and equipment requirements are not currently considered in the decomposition process.

In a segmented scheduling scenario, the mission is broken down into $k = 1, \dots, K$ non-overlapping segments of segment length, T (i.e., $KT = \text{Mission Length}$). Assuming equal resource and crew availability within k , the total availability of crew member g is $C_{kg} = C_g / K$; likewise, the total availability of resource type b is $R_{kb} = R_b / K$. (If the actual crew and resource availabilities within each segment can be found, this approximation is not required). Crew over-subscription, α , is found over the entire set of crew members since, due to crew flexibility in model requirements, it is difficult to find the exact level for each member. The over-subscription of each resource type b is given by α_b . These quantities are given by

$$\alpha_b = \frac{\sum_{i=1}^M N_i \text{rmin}_{ib}}{\sum_{k=1}^K R_{kb}}, \text{ for resource type } b$$

and

$$\alpha = \frac{\sum_{i=1}^M N_i NC_i cmin_i}{\sum_{g=1}^G \sum_{k=1}^K C_{kg}}.$$

Using α and α_b , the total resource and crew availability within a segment are adjusted as per the level of over-subscription. Consequently, more models would be assigned to segments than can actually be scheduled. The resource and crew availabilities found are aggregate, and do not reflect the specific values of these quantities at specific times. If $Pmin_i > T$ for some i , that model must be split into sub-models which must be constrained to be assigned to adjoining segments.

Mathematical Model for Assignment

Specifying the macrowindows of a model as segments in which the model is to be attempted (i.e., temporal decomposition) is equivalent to assigning model performances to timeline segments. The objective of temporal decomposition is to allocate model performances to appropriate segments such that they have adequate temporal opportunity for being scheduled; the assignment should enable the scheduling of as many models as possible. The preference of model i for being assigned to segment k is given by O_{ik} , and w_i is its weight, or importance. The relational, sequencing, and "soft" constraints are not included in this formulation. We assume that these can be suitably incorporated in the preferences, O_{ik} , through a

preference-setting module. Since a model may start in one of several possible segments, the preference for assigning a model to a segment can be naturally expressed in a fuzzy manner. The total preference for assigning (i.e., scheduling) a model in a segment can be found by rating only one "attribute" at a time (see Badran, 1988 and Kacprzyk and Fedrizzi, 1988).

Each model i may have $v = 1, \dots, V$ time windows of the form $\{Wmin_{iv}, Wmax_{iv}\}$, within which N_{iv} performances are to be scheduled. The number of performances is generally over-requested, that is

$$\sum_{v=1}^V N_{iv}.$$

The minimum and maximum windows of i are $Wmin_i = \min_v \{Wmin_{iv}, \text{ for } 1 \leq v \leq V\}$ and $Wmax_i = \max_v \{Wmax_{iv}, \text{ for } 1 \leq v \leq V\}$. These are transformed to $L_i = [Wmin_i / T]$ and $U_i = [Wmax_i / T]^+$, which represent the first and last segment in which i may be scheduled. (The notation " $[X]$ " and " $[X]^+$ " indicates that X is rounded down, and up, respectively). For example, if the time window of model i is the entire mission, then $L_i = 1$ and $U_i = K$, where K is the last segment of the mission. If i cannot be scheduled in segment k , say, due to intermittent windows, then $O_{ik} = 0$. Time windows are modeled only implicitly; only the segments for an activity, and not the specific times, are modeled. The number of performances of i that may be assigned to k is limited by the performance duration and minimum performance delays. The maximum number of performances of model i that may be

assigned to segment k is given by $MAX_{ik} = [T / (Pmin_i + Dmin_i)]$. Also, the performances of certain models are to be distributed over the timeline. For instance, photographic experiments on planetary targets are often required to be evenly spaced over the mission in order to enhance their scientific return.

Based upon the above notation, the following mathematical formulation for the assignment problem has been developed:

$$\text{Maximize } \sum_{i=1}^M \sum_{k=L_i}^{U_i} w_i O_{ik} X_{ik} \quad (1)$$

Subject to:

$$\sum_{k=L_i}^{U_i} X_{ik} \leq N_i, \quad \text{for all } i, \quad (2)$$

$$X_{ik} \leq MAX_{ik}, \quad \text{for all } i \text{ and for } k = L_i, \dots, U_i, \quad (3)$$

$$\sum_{i=1}^M r_{ib} X_{ik} \leq \alpha_b R_{kb}, \quad \text{for all } k \text{ and } b, \quad (4)$$

$$\sum_{i=1}^M c_i C_{ig} X_{ik} \leq \alpha C_{kg}, \quad \text{for all } k \text{ and } g, \quad (5)$$

$$\sum_{g \in Cset_i} C_{ig} = NC_i, \quad \text{for all } i, \quad (6)$$

$$X_{ik} \geq 0, \quad X_{ik} \text{ is integer, and } C_{ig} \in \{0,1\}.$$

The objective, (1), is to assign model performances to segments so as to maximize the total preference of the assignment; (2) states that, for each model, the requested number of performances should be assigned to the segments with respect to the time window constraints, while (3) says that no more than a specified maximum number of performances of activity i should be assigned to segment k ; (4) and (5) state that the total amount of resource consumption and crew utilization, respectively, must not exceed the total availability in a segment for the different resources and crew members (the required amount of resources and crew time should be allotted to each performance); (6) says that only the required number of crew members should be used by an activity. The X_{ik} 's are integer (non-negative) decision variables denoting the number of performances of model i assigned to segment k . The C_{ig} 's are zero-one variables specifying whether or not crew member g was utilized for activity i .

The number of X_{ik} (decision) variables, $\sum_{i=1}^M (U_i - L_i + 1)$, depends on the time window of models, and the number of C_{ig} variables, $\sum_{i=1}^M n(Cset_i)$, depends on the set of crew members, $n(Cset_i)$, specified by the models. The model requires a total of $[2M + \sum_{i=1}^M (U_i - L_i + 1) + K(B + G)]$ constraints (excluding integrality and non-negativity). Certain resource types b may be overlooked if $\sum_{i=1}^M N_i r_{ib} \leq \sum_{k=1}^K R_{kb}$. Only the most constraining resources need be considered. Likewise, if crew members are not a constraining resource, then constraints (5) and (6) can be removed, which would make the formulation more tractable. Since time is considered only implicitly, the model can be readily extended to long missions. The formulation is not affected by

the mission or segment lengths, but rather by the number of segments. However, (non-linear) integer programming models are generally difficult to solve and their practical use is limited due to high computational requirements. Due to this difficulty, heuristic methods that provide a "good" solution in a reasonable amount of time are often employed. In the following chapter, we present one such procedure for assigning model performances to segments.

CHAPTER IV

HEURISTIC ASSIGNMENT PROCEDURE FOR TEMPORAL DECOMPOSITION

Assignment Heuristic

A model having no predecessor is eligible for selection immediately. Also, a model is eligible for selection if at least one performance of all of its predecessors has been scheduled. These facts are quite obvious, and have been used widely in scheduling problems that involve precedence constraints (e.g., see Kelley, 1963). In the heuristic given below, one model is selected at a time from a set of eligible models, E , and its performances are assigned to appropriate segments in a sequential manner. The assignment heuristic is basically an approximation of a scheduling process, and may lead to a balanced assignment owing to the flexibility in assigning individual activities, rather than clusters of activities. Any concurrency and sequencing requirements specified by the steps are viewed as mandatory restrictions on the model as a whole. Accordingly, a model desiring concurrence is attempted in the same segment as that of its required model, which may enable the steps to be scheduled together. Any sequencing delays between models are to be considered only while scheduling (i.e., by ESP).

Let i_p and i_c be the set of predecessors and concurrencies, respectively, of model i , and i_{ca} be the subset of i_c that has already been assigned. Let ST_i and CT_i denote the start time and completion time, respectively, of the first performance of

model i . A set of preferred segments, PS_i , is used to indicate the segments (i.e., times) at which model i prefers to be scheduled, and to skip unfit increments within its time window. If both concurrencies and segment preferences exist for a model, the former are given priority. PT is a variable that indicates the present time for the selected model. $N_{i[PT]}$ is the minimum number of preferences of i that can be assigned to the segment $[PT / T]$ with respect to the resource, crew, and temporal constraints. A_{ik} is the number of performances of i that are actually assigned to segment k , and A_i is the total number of performances of model i assigned to all segments. The unassigned model performances, UA_i , are assigned "evenly" over the segments of the model (this procedure is described in some detail in the next section).

Step 0. Initialize. $A_{jk} = 0$, for all j , and for $k = L_i, \dots, U_i$.

Step 1. Find the set of eligible models, E .

$$E = \{u: u_p = 0 | u = 1, \dots, M\} \text{ and } \\ \{u: A_j > 0, \text{ for all } j \in u_p | u = 1, \dots, M\}.$$

Step 2. The select and assign process is repeated until all eligible models have been attempted. If $E = 0$, END.

Step 3. Select a model i using some selection rule, π . $i = j: \min_{j \in E} \pi^j$.

Step 4. Find the "earliest" start time of i with regard to its time windows, predecessors, concurrencies, and preferred segments. The earliest start time (segment) of a model is the maximum of (a) the minimum performance time

window, and (b) the maximum of the completion time of the first performance of its predecessors, if any. If possible, the model is made to start in the same segment as the first performance of any concurrent model, in order to provide them a chance for being performed concurrently. Consequently, even if the first performance of the models are not scheduled together, concurrency may still be met with some other performance.

$$ST_{i,ip} = \max_u \{CT_u | u \in i_p\}; ST_{i,ica} = \min_u \{ST_u | u \in i_{ca}\}.$$

(If $i_p = 0$, $ST_{i,ip} = Wmin_i$; likewise, if i_{ca} or $i_c = 0$, $ST_{i,ica} = Wmin_i$).

$PT = Wmin_i$; If $ST_{i,ip} > PT$, $PT = ST_{i,ip}$; If $[ST_{i,ica} / T]^+ > [PT / T]^+$,

$$PT = [ST_{i,ica} / T]^+ \times T.$$

If $\min_{k \in PS_i} k > [PT / T]^+$, $PT = k \times T$; Else, update PS_i to point to segment

k : $k \geq [PT / T]^+ + 1$. Skip this sub-step if $PS_i = 0$.

Step 5. Model i is to be attempted serially over the segments until all of its performances have been assigned, or until the maximum time window is reached. Step 5 ensures that no model performance is assigned outside its time window.

If $PT \geq U_i \times T$, $UA_i = N_i - A_i$; Go to 1.

Step 6. Find the minimum number of performances that may be assigned to segment $[PT / T]^+$, subject to the availability of sufficient duration, resources, and crew time.

$$N_{i[PT]} = \min \{ N_{i[PT]T}, N_{i[PT]R}, N_{i[PT]C} \}, \text{ where}$$

$$N_{i[PT]T} = [([PT]^+ - PT) / (Pmin_i + Dmin_i)]^-,$$

$$N_{i[PT]R} = \min_b [R_{[PT]b} / rmin_{ib}]^-,$$

$$N_{i[PT]C} = \min_{g \in CC[PT]} [C_{[PT]g} / cmin_{ig}]^-,$$

and $CC[PT] \in Cset_i$ is the set of crew members who have been utilized minimally in segment $[PT / T]^+$; $n(Cset_i) \geq n(CC[PT]) = NC_i$.

Step 7. If no performances can be assigned in $[PT / T]^+$, try the next possible, or next preferred, segment.

If $N_{i[PT]} < 1$, $PT = ([PT / T]^+ + 1) \times T$, or, if PS_i is not equal to 0, then $PT = PS_i \times T$; Go to 5.

Step 8. Find $A_{i[PT]}$, such that $A_i \leq N_i$; the number of performances assigned must not exceed MAX_{ik} . The resulting performances are relegated to segment $[PT / T]^+$, and the resource and crew availabilities are suitably updated. Crew flexibility is exploited to engage the least-utilized members. The actual start time and completion time of the first performance are noted; these are used only as indicator variables in the assignment, and are irrelevant while scheduling.

If $A_i = 0$, $ST_i = PT$ and $CT_i = PT + pmin_i$.

If $N_{i[PT]} > MAX_{ik}$, $N_{i[PT]} = MAX_{ik}$.

If $N_i - \sum_{k=L_i}^{([PT]^+-1)} A_{ik} > N_{i[PT]}$, then $A_{i[PT]} = N_{i[PT]}$; else, $A_{i[PT]} = N_i - \sum_{k=L_i}^{([PT]^+-1)} A_{ik}$.

$$R_{[PT]b} = R_{[PT]b} - A_{i[PT]} \times rmin_{ib}, \text{ for all } b,$$

$$C_{[PT]g} = C_{[PT]g} - A_{i[PT]} \times cmin_i, \text{ for all } g \in CC_{[PT]}.$$

Step 9. If some performances are still unassigned, try the next (preferred) segment; else, in the next step, determine if any new model of those previously ineligible can now be released into the selection stage due to the previous allocation.

$$([PT]^{-1})$$
 If $N_i - \sum_{k=L_i} A_{ik} > 0$, $PT = ([PT / T]^{-1} + 1) \times T$, or, if PS_i is not equal to 0,
 then $PT = PS_i \times T$; Go to 5; else, Go to 1.

Activity Selection Rules

The following four selection rules were used in the heuristic to select a model $i \in E$ with the :

- (1) fewest number of requested performances, ($\pi_{np,j} = N_j$, for all j),
- (2) shortest activity duration, ($\pi_{ad,j} = N_j (Pmin_j + Dmin_j)$, for all j),
- (3) shortest time window, ($\pi_{tw,j} = Wmax_j - Wmin_j$, for all j),
- (4) highest criticality, ($\pi_{tc,j} = (\pi_{tw,j} - \pi_{ad,j}) / \pi_{tw,j}$, for all j).

A fifth rule, π_r , was used to select i randomly. For all the rules, ties are resolved in favor of the model that has the most number of successors. Several composite (bi-level) selection rules may also be employed. For example, models can be grouped

based on π_{np} or π_{tw} , and a second rule (e.g., use of similar resources) could be applied to discern between models in the same group.

A sixth rule, π_e (the "even" heuristic), assigns the performances of model i "evenly" over the possible segments, L_i to U_i , ignoring all other constraints. The number of performances, A_{ik} , of model i assigned to segment k depends only upon its time window, $\{Wmin_i, Wmax_i\}$. If $N_i = 1$, then this single performance is assigned to the first (preferred) segment of i . If $N_i < (U_i - L_i + 1)$, the performances are assigned to the preferred segments, and earlier segments. In general, with this "even" heuristic, the performances of model i are distributed as follows:

$$N_{iL_i} = [\{(L_i + 1)T - Wmin_i\} / \pi_{tw,i} \times N_i]^+, \text{ for all } i.$$

$$\text{If } N_{iL_i} \geq N_i, A_{iL_i} = N_i; \text{ Else, } A_{iL_i} = N_{iL_i}, \text{ for all } i.$$

For $i = 1, \dots, M$

For $k = (L_i + 1), \dots, (U_i - 1)$

$$N_{ik} = [T / \pi_{tw,i} \times N_i]^+$$

$$\text{If } N_{ik} \geq N_i - \sum_{w=L_i}^{k-1} A_{iw}, A_{ik} = N_i - \sum_{w=L_i}^{k-1} A_{iw}; \text{ Else, } A_{ik} = N_{ik}$$

Next k

Next i

$$N_{iU_i} = [(U_i T - Wmax_i) / \pi_{tw,i} \times N_i]^+, \text{ for all } i.$$

$$\text{If } N_{iU_i} \geq N_i - \sum_{w=L_i}^{U_i-1} A_{iw}, A_{iU_i} = N_i - \sum_{w=L_i}^{U_i-1} A_{iw}; \text{ Else, } A_{iU_i} = N_{iU_i}, \text{ for all } i.$$

The following factors result in a loosely-linked decomposition, and, thus, deter the independent (parallel) scheduling of the time increments:

- Model performances must slide over the time continuum. Thus, it is imperative to schedule certain performances in a split fashion (partially). In fact, models whose performance duration are greater than the segment length (i.e., $P_{min_i} > T$) must be scheduled in a split fashion. When splitting, the remaining duration (i.e., unloaded steps) must be carried over to the subsequent segment(s). The capability to schedule a model performance partially is currently not available in ESP.
- The required delay between performances or steps, and any sequencing delays between models, should be satisfied. The scheduler should consider such delays with respect to models assigned in preceding segments.
- The scheduled duration is likely to be higher than the minimum duration. In fact, ESP tries to maximize step durations, thus maximizing performance duration. Model performances not scheduled in the segment to which they were originally assigned may be schedulable in others. Thus, after scheduling a segment, the unscheduled performances must be moved to the next possible increment. Some model performances may not be scheduled at all due to resource over-subscription.

CHAPTER V

EXPERIMENTAL RESULTS

Introduction

In this chapter, results of experimentation on the use of segmented scheduling and temporal decomposition are reported. Based on whether the mission is scheduled in a segmented (SS) or non-segmented (NS) manner, and based on whether the activities are temporally decomposed (TD) or non-decomposed (ND), each mission can be scheduled in four different ways, namely ND/NS, ND/SS, TD/NS, and TD/SS.

NS corresponds to scheduling a mission fully, rather than in separate segments. ND means that macrowindows are not used to divide the mission into time segments for the purposes of assigning model performances. ND/NS is the way NASA generally does mission scheduling. In this research, we are primarily concerned with the effectiveness of ND/SS and TD/SS schedules. In the former, we examine the effects of scheduling missions in a segmented manner only; in the latter, we investigate the usefulness of both restricting activities within segments (TD) *and* scheduling the segments incrementally (SS). TD/NS is examined basically to gain some understanding as to how good we can do by restricting (only) the activities, and still fully scheduling the mission. TD/NS can be viewed as a sort of loose upper bound on the performance of TD/SS.

Activity data for two hypothetical missions was made available for use in this study. These missions include a small Space Station mission with a 48-day duration and an 8-day Spacelab mission. For the ND/SS and TD/SS schedules, both missions are split into eight segments. For the Space Station, the segment length, T , equals six days, while T equals 1 day for the Spacelab mission. Results for these missions are presented in Tables 1 and 2 below. For all these schedules, the weighting factors in ESP's schedule grade function were set equal to one.

In the tables, "GM" is the grade of the schedule obtained using the grade maximization rule. "RB" is the schedule that had the best grade value among five randomly-generated schedules. For each mission, 15 different sequences of model groups were first tested using ND/NS (GM). In all the results given below (for both missions), the sequence that gave the best GM value was used. Typically, there is only a slight difference (one or two grade points) between GM and RB, and GM takes much less time. Macrowindows were not defined for models for which $P_{min} > T$ and, since ESP does not have the capability of scheduling performances partially over the segments, these were not scheduled at all in the ND/SS, TD/NS, and TD/SS schedules.

In Tables 1 and 2, Γ is the grade value, and TP and TM denote the total number of performances and models, respectively, which were scheduled. TC and TA denote, respectively, the crew time and exposure/activity time scheduled (in hours). The last column is the CPU time (in seconds) taken by ESP for scheduling

the mission on a VAXstation 4000 Model 60. "Rule" denotes the selection rule used for temporal decomposition. The assignment and "even" heuristic were coded in "C", and the CPU times taken (in seconds) on a SUN SPARCstation 1+ to decompose the Space Station and Spacelab missions are denoted by "CPU" in the "Rule" column. The first two rows in the tables give the details of the timeline that is imported before beginning each scheduling session. This timeline consists of the crew and system operations which are hard-scheduled by NASA (e.g., crew sleep schedules). The next row gives the total mission request.

When using the assignment heuristic, the unassigned performances, UA_i , are allocated evenly over the model's segments. If $UA_i = 1$, this lone performance is assigned to the first (preferred) segment. Even allocation would make the assignments from the different rules (via the heuristic) be fairly similar. This is reflected in the TD/SS grades of both missions, which are nearly the same, irrespective of the rule employed.

Space Station Mission Results

Results for the Space Station mission are given in Table 1. By comparing ND/NS (NASA's general scheduling procedure) with TD/NS, we see that we lose about two grade points by defining the macrowindows on models (i.e., "TD") - fewer performances are scheduled in the TD/NS schedules. While the CPU times for ND/SS and TD/SS schedules are quite low, their performance is fairly poor when

Table 1 - Results for Space Station Mission

		Γ	TP	TM	TC	TA	CPU
Rule (CPU')	Import Request	-	200	40	7635	3938	-
	Import TL	98.8	193	40	7464	3955	-
	Mission Request	-	1387	83	7903	20752	-
	ND/NS GM	95.5	1292	81	7660	19972	599
	RB	96.3	1313	82	7662	20145	872
	ND/SS GM	84.4	1282	71	7654	13646	320
	RB	84.6	1303	71	7656	13657	971
π_r (0.3)	TD/NS GM	93.7	1265	79	7652	19727	832
	RB	94.4	1276	80	7636	19897	1073
	TD/SS GM	82.7	1256	69	7637	13553	266
	RB	82.7	1270	69	7626	13500	444
π_{tc} (0.3)	TD/NS GM	93.4	1240	79	7601	19944	633
	RB	93.7	1243	80	7603	19888	960
	TD/SS GM	82.6	1244	70	7596	13593	222
	RB	82.3	1245	70	7590	13522	647
π_{tw} (0.2)	TD/NS GM	93.7	1263	79	7636	19821	540
	RB	94.5	1268	81	7635	19945	861
	TD/SS GM	83.4	1258	70	7640	13663	173
	RB	83.3	1267	70	7633	13644	651
π_{ad} (0.2)	TD/NS GM	92.1	1230	79	7613	19085	736
	RB	93.0	1236	81	7615	19235	937
	TD/SS GM	81.7	1221	70	7608	12916	199
	RB	81.5	1231	70	7603	12901	569
π_{np} (0.3)	TD/NS GM	93.8	1253	80	7621	19800	757
	RB	93.8	1268	80	7626	19726	934
	TD/SS GM	82.8	1243	70	7616	13446	231
	RB	82.3	1257	69	7611	13442	613
π_e (0.1)	TD/NS GM	93.5	1255	79	7657	19644	495
	RB	94.3	1282	80	7652	19783	722
	TD/SS GM	83.3	1249	70	7646	13649	236
	RB	83.6	1278	70	7654	13659	542

compared with that of ND/NS (the difference in grade is over ten points). The CPU time differences may be much more significant for very long missions.

The TD/SS schedules take much less time for all the cases, and this difference would be significant for long missions. The primary difference between TD/NS and TD/SS for almost all the rules is that some activities having a high activity (exposure) time do not get scheduled using TD/SS. These are probably the activities whose $P_{min_i} > T$. Also, ND/SS is only slightly better than the TD/SS's. But the TD/SS's take much less time than ND/SS. For long missions, using TD might be helpful because of this fact.

It appears that the six assignment rules can be divided into three groups with respect to their performance for this problem. The π_{tw} (i.e., shortest time window) and π_e (i.e., "even" decomposition) rules appear to perform best for the TD/SS cases. The π_{np} (i.e., fewest number of requested performances), π_{tc} (i.e., highest criticality), and π_r (i.e., random) rules appear to perform equivalently to π_{tw} and π_e for the TD/NS cases, but slightly worse for TD/SS. Finally, the π_{ad} (i.e., shortest activity duration) rule appears to be the worst rule for both TD/SS and TD/NS.

Spacelab Mission Results

Results for the Spacelab mission are given in Table 2. For this mission, there is very little difference between ND/NS and ND/SS. Also, none of the TD/NS or TD/SS cases match the ND/SS value. So, once again, doing segmented scheduling

Table 2 - Results for Spacelab Mission

		Γ	TP	TM	TC	TA	CPU
Rule (CPU')	Import Request	-	771	160	760	2664	-
	Import TL	99.0	392	160	840	3573	-
	Mission Request	-	991	173	772	2708	-
	ND/NS GM	110.4	764	173	1015	3752	311
	RB	110.5	768	173	1014	3752	479
	ND/SS GM	110.3	767	172	1017	3750	106
	RB	110.2	769	172	1012	3749	759
π_r (0.4)	TD/NS GM	104.6	715	171	876	3702	122
	RB	104.6	715	171	876	3702	253
	TD/SS GM	105.0	733	171	874	3712	136
	RB	105.1	734	171	875	3714	576
π_{tc} (0.4)	TD/NS GM	104.9	721	171	881	3702	150
	RB	104.9	721	171	880	3702	182
	TD/SS GM	104.9	721	171	881	3701	143
	RB	104.9	721	171	880	3701	545
π_{tw} (0.4)	TD/NS GM	105.2	721	171	892	3707	139
	RB	105.3	722	171	894	3709	179
	TD/SS GM	105.2	721	171	892	3706	164
	RB	105.2	721	171	891	3705	460
π_{ad} (0.5)	TD/NS GM	104.7	716	171	876	3704	166
	RB	104.7	716	171	875	3704	236
	TD/SS GM	104.4	712	171	873	3702	153
	RB	104.5	713	171	874	3704	521
π_{np} (0.3)	TD/NS GM	104.5	713	171	874	3704	144
	RB	104.6	714	171	876	3704	193
	TD/SS GM	104.5	713	171	876	3702	157
	RB	104.6	714	171	876	3704	574
π_c (0.2)	TD/NS GM	105.3	736	171	876	3710	158
	RB	105.4	737	171	876	3713	221
	TD/SS GM	105.3	736	171	876	3710	166
	RB	105.3	736	171	876	3711	476

alone (i.e., without temporal decomposition) seems to be a good option.

Interestingly, there is very little difference between TD/NS and TD/SS for most of the rules, except for the Random rule, where some extra performances are scheduled for TD/SS. It is not clear why none of the TD/NS cases are as good as ND/SS.

With regard to TD, the π_{tw} and π_c rules again appear to perform best.

For this mission, there seems to be no real advantage in using TD in terms of the scheduling time since ND/SS (GM) actually takes less time than the TD/SS's. Also, comparing the CPU times of TD/NS and TD/SS shows that the former actually takes less time, even though the mission is scheduled completely. In fact, the ND/NS and ND/SS yield much better grades even though their CPU times are only slightly longer. However, it should be noted that this mission is probably not long enough to serve as a good test for segmented scheduling and decomposition. Most of its activities are quite short, also. To adequately compare the effectiveness of ND/SS with TD/SS, we may need to have fairly long missions so that the results can be extrapolated to durations expected in Space Station missions.

General Comments

Models not scheduled in a segment were not transferred to the next possible segment, since such a mechanism is not currently available. The schedule grades obtained by way of decomposition can be improved by scheduling certain models partially within the segments, and by transferring unscheduled performances to the

next possible segment. The lack of a long-duration test case makes it difficult to fairly evaluate the methods. Further experimentation is needed in order to make more general statements about the usefulness of the proposed methods. However, it should be noted that the π_{tw} and π_e rules, and the ND/SS method, are simple, realistic, and appear to work relatively well.

The Space Station mission has little concurrency, activity sequencing requirements, or energy requirements. It does, however, have some long activities. On the other hand, the Spacelab mission has a great deal of sequencing, concurrency, and energy requirements, but with fewer long models. Realistic mission scheduling problems tend to involve many activities, with multiple objectives and numerous constraints. Knowledge of the critical characteristics for a particular mission should be very helpful in identifying appropriate selection rules and decomposition methods. There would probably be some advantages in categorizing the characteristics of a mission, at least in some aggregate sense, since this should provide some insight into the likely usefulness of segmented scheduling and decomposition for that particular mission. Differences in mission characteristics are likely to lead to substantial differences in the quality of schedules produced by the different methods.

There are several possible means of handling very long activities (i.e., those with $P_{min_i} > T$). They could be split into sub-models (or steps) which have sequencing constraints between them so that the sub-models are scheduled contiguously. Alternatively, these activities could be partially scheduled, with the

remaining processing time carried over into the next time segment. However, if these activities have intermittent time windows, the partial scheduling approach may become quite complicated. It could be possible to use the step delays and performance delays to account for intermittent time windows. However, a representation scheme which does not involve such activity splitting would be much more efficient and desirable, given that the performance duration of many models in a space station mission are expected to last as long as several weeks (Stacy and Jaap, 1988).

CHAPTER VI

SUGGESTIONS FOR FUTURE RESEARCH*

General Comments

Both ESP and the decomposition algorithm employ the select and assign framework. ESP's loading algorithm attempts to schedule one model performance at a time. The assignment algorithm, however, attempts to assign all model performances to suitable segments sequentially, so as to preserve the continuity of placing (scheduling) model performances over the time segments. Clearly, the idea behind the heuristics is quite simple. It remains to be seen how effectively more sophisticated procedures may perform in temporally decomposing the activities. If activities in a mission are homogeneous, there must be some advantage in attempting clustering-based methods for decomposition. However, in general, there is no obvious way to cluster the activities due to the variety of different activity types common in space missions.

An alternative strategy would be to attempt model performances in only one segment at a time. Assignment progresses by filling the segments sequentially, rather than assigning a model serially. If no model can be assigned to a segment, allocation

* Venkata R. Neppalli served as a co-author for this chapter.

proceeds to the next segment. The time window and time criticality rules can be implemented in a dynamic fashion wherein only the active segments are considered while finding the time available for a model. Resource-based selection rules can be employed (for instance, "select the model that will maximize resource utilization if scheduled"). Special consideration may be required to account for the delay between performances. Model performances should be assigned in a global sense since an optimal assignment in the individual segments does not lead to an optimal overall solution (Sadowski and Jacobson, 1978). If a global assignment strategy can be identified to perform decompositions of this kind, the assignment process can focus on allotting the most suitable models to a given segment rather than having to allocate models far into the future.

While performing segmented scheduling, it may be necessary to use different scheduling rules in the various segments. A proficient scheduler may be able to identify the most likely set of rules that might yield a good schedule. Also, adaptively switching between a set of rules during the scheduling process may be employed; this approach has yielded better results than using a single dispatching rule, in a production scheduling environment (for instance, see Chandra and Talavage, 1991). Inductive learning techniques (such as Genetic Algorithms, or ID3) can be used to categorize problem situations and to identify effective rules for these situations based on their performance. Such a switching mechanism will also be very

useful in the assignment (decomposition) process since schedule quality depends heavily on the order of selection of tasks.

The evaluation function for the decomposition process should judge if the scientific and operational requirements can be met by a given assignment, while scheduling. A fuzzy preference-setting module or a method similar to the "suitability function" scheme used in SPIKE may be necessary. A suitable strategy while performing segmented scheduling would be to identify good partial schedules, in the early segments, which may serve to yield reasonably "good" timelines. Evaluation of the quality of a partial schedule is a key issue as this may enable an efficient distinction between the various assignment configurations without actually scheduling them completely. However, this may be intractable due to resource assignment conflicts, resource over-subscription, stochasticity, etc. The schedule grade function only measures the aggregate quality of a schedule, and does not consider the priority/importance of activities, the extent to which important constraints are satisfied, etc. In this regard, it might be useful to have two grade functions - an aggregate one to distinguish between the poor and good schedules, and another to be used in evaluating schedules with good aggregate grade values.

Models whose performance duration is greater than the time length of a segment can be divided into sub-models. However, a representation scheme which does not involve such splitting would be more efficient and desirable given that the

performance duration of (many) models in a Space Station mission is expected to be several weeks.

Rescheduling and schedule repair are likely to be important tasks as schedule deviations in the earlier segments may cause a compounding ("ripple") effect in the subsequent segments, thus complicating the scheduling problem. In such a situation, it would be necessary to preserve the temporal position of certain high priority activities which may be critical to the mission's success. Schedule revision must focus on preserving such activities at the expense of low priority activities.

Possible Use of Artificial Neural Networks

In view of the inherent complexity of space mission scheduling, decomposition is viable and important. The preliminary results presented herein show promise with respect to temporal decomposition, but are disappointing with respect to segmented scheduling. Several factors, such as the complexity and size of the problem, selection bias, and so on, may have contributed to these results. At any rate, it appears that more sophisticated methods for temporal decomposition and segmented scheduling should be investigated. Even though dispatching rules such as those used in this study provide simple means of accomplishing activity selection, more sophisticated approaches may handle the problem more effectively.

Approaches which use adaptive learning to exploit the problem structure may be considered to extend the present solution framework. We consider neural

networks as a promising technique, and propose the following framework which may provide guidance for evolving more sophisticated approaches for temporal decomposition with the capabilities of adaptivity and learning. In the proposed approach, neural networks may be incorporated as a component for assessing and learning the schedule quality and characteristics. The proposed approach may provide a better alternative for doing temporal decomposition without much degradation in schedule quality as compared to the methods currently in use.

Neural networks model the human nervous system and have been successfully applied to several classification and clustering problems which are as complicated as natural language processing problems. The suitability of neural networks can be justified by their speed and ability to learn the problem characteristics in an unsupervised manner.

Several factors influence the efficiency of decomposition, and in an ideal case the approach should be able to decompose the problem into "disjoint" sub-problems. However, in many problems this may not be possible. In order to deal with such problems, which result in inter-connected inter-dependent sub-problems, an approximate decomposition must be used. Also, decomposability of the problem, combined with the optimality criteria, will affect the performance of the decomposition approach in terms of efficiency and feasibility of the final schedule. In the proposed approach using neural networks, an iterative decomposition of the timeline can be considered, and a feed-forward neural network can be used to

adaptively classify the activities into segments in order to achieve a quality schedule. In other words, the proposed approach investigates the means of replacing the dispatching rules with a neural network, hence relieving the burden of understanding and optimizing the bias in the selection methods.

In this approach, the problem segments are iteratively loaded with the activities. At the present time, the facility of using an external program to submit the activities and run the ESP is unavailable. Therefore, instead of actually scheduling the activities in each segment, an approximation of the actual scheduling is used to estimate the performance of the network. Due to the iterative loading of activities in each segment, the approach considers sequencing as well as relational constraints.

In the proposed approach, each performance of an activity will be considered as an entity and the problem consists of forming a dynamic and iterative classification network which will be used to evolve an approximate schedule by classifying the entities of the problem. The network basically consists of two sets of input nodes. The first set will be used to input the attributes of the segment and the second set will be used to input the attributes of the activities.

As mentioned above, we assume that the timeline is decomposed into suitable segments. Segments are then considered one at a time. From the basic set of eligible activities, each activity is fed into the network to decide whether the activity belongs to the segment or not. Hence, the output from the network, from a single output node or a set of output nodes, is used to determine whether the activity belongs to the

present segment or not. Once an activity is assigned to the present segment, the attributes of the segment are changed by considering the consumption of resources by the assigned activity. Hence, before loading the next activity, the attributes of the segment are updated. This implies that the network evaluates the suitability of the present activity to the current segment. In other words, the approach basically forces the network to form clusters in each segment.

After finding the set of activities which are assigned to the current segment, the procedure continues to the next segment, and so on. After completing all the segments, an approximate loading algorithm is used to schedule the activities in order to empirically estimate the grade. It should be noted that the grade estimation of a schedule is an approximation and is expected to reflect the actual grade. Using this measure, the feed-forward neural network adjusts its weight in order to enhance its performance measure. The procedure is repeated until a desired performance level is achieved.

The proposed framework employs a structured approach and provides a means of iterative decomposition. The performance of the approach depends on several factors such as (i) defining the attributes of an activity, (ii) defining the attributes of a segment, (iii) the procedure for updating the attributes of a segment, (iv) the procedure for approximating and evaluating the schedule, and (v) the architecture and type of the network.

A neural network approach to the problem offers several potential advantages. These include flexibility in incorporating user-defined selection bias, the provision of a means of analyzing and estimating the important attributes of activities and segments (and thereby deriving a good schedule), the offline nature of the procedure, and the ease with which the method can be parallelized.

Obviously, several important issues must be resolved in order to implement the proposed neural network framework. Also, several possible means of implementing the framework need to be investigated to determine the best design of such an approach. Two possible implementation approaches include (i) using a parallel network architecture and assigning an individual network to each segment, with all the individual networks connected in parallel, and (ii) extending to a parallel distributed network in order to process all the segments simultaneously.

CHAPTER VII

CONCLUSIONS

In the course of conducting this research, a thorough review of the literature pertaining to space mission scheduling problems has been conducted. The unique features of such problems have been highlighted. The need for segmenting long-duration problems for purposes of activity assignment and detailed scheduling (i.e., "temporal decomposition" and "segmented scheduling", respectively) has been documented. The problems inherent in attempts to perform temporal decomposition and segmented scheduling have been discussed. A non-linear, zero-one integer programming formulation has been presented as one means of defining the nature of the temporal decomposition problem.

Due to the computational complexity of the temporal decomposition problem, a heuristic assignment framework is presented, and implemented using several different simple activity selection rules. All combinations of segmented vs. non-segmented, and decomposed vs. non-decomposed techniques were tested using data from one sample Space Station mission and one sample Spacelab mission. These preliminary results indicate that (i) using segmented scheduling, rather than non-segmented scheduling, may or may not result in a degradation in the quality of the schedule, depending on the characteristics of the mission involved, (ii) the relative performance of decomposition vs. non-decomposition also appears to be mission-dependent, (iii)

two of the activity selection rules used within the heuristic appeared to perform best across all experimental conditions, namely, the selection of activities with the shortest time window first (i.e., " π_{tw} "), and the assignment of performances of a model evenly across its possible segments (i.e., " π_e ").

As is often the case with preliminary research, numerous questions remain to be studied. The results of the experimental analysis clearly indicate the need for means of defining and classifying the characteristics of a specific mission, and understanding how those characteristics affect the quality of schedules produced by the use of temporal decomposition and/or segmented scheduling. An offline learning technique, such as neural networks, may be useful in classifying missions for this purpose. The use of clustering approaches, in general, for this type of problem deserves further attention. The use of adaptive selection rules should also be studied, as well as means for identifying "good" partial schedules as the schedules are being developed. Finally, the issues of rescheduling and schedule repair are suggested as critical areas of future research on the planning of long-duration missions.

REFERENCES

- Badran, Y. (1988), "Preference Ranking of Discrete Multi-attribute Instances," *European Journal of Operational Research*, Volume 34, pp. 308-315.
- Baker, K. R. (1974), *Introduction to Sequencing and Scheduling*, John Wiley, New York.
- Blackstone, J. H., D. T. Phillips, and Hogg, G. L. (1982), "A State-of-the-art Survey of Dispatching Rules for Manufacturing Job Shop Operations," *International Journal of Production Research*, Volume 20, pp. 27-45.
- Boctor, F. F. (1990), "Some Efficient Multi-heuristic Procedures for Resource-constrained Project Scheduling," *European Journal of Operational Research*, Volume 49, pp. 3-13.
- Bullington, S. F. and J. P. Jaap (1992), "A Comparison of Space Mission Activity Scheduling Problems with Production Scheduling Problems," unpublished manuscript.
- Chandra, J. and J. Talavage (1991), "Intelligent Dispatching for Flexible Manufacturing," *International Journal of Production Research*, Volume 29, pp. 2259-2278.
- Chrysosolouris G., J. Pierce and Dicke K. (1991), "An Approach for Allocating Manufacturing Resources to Production Tasks," *Journal of Manufacturing Systems*, Volume 10, No. 5, pp. 368-382.
- Chu, C., M. C. Portmann, and Proth, J. M. (1992), "A Splitting-up Approach to Simplify Job-shop Scheduling Problems," *International Journal of Production Research*, Volume 30, No. 4, pp. 859-870.
- Garey, M. R. and D. S. Johnson (1979), *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco.
- Gaspin, C. (1989), "Mission Scheduling," *Proceedings of the 1989 Goddard Conference on Space Applications of Artificial Intelligence*, Maryland, pp. 75-86.

Goldin, D. S. (1993), "Redesign Team Presents Viable Options; Achieves Significant Savings," *Station Break*, Volume 5, No. 4, pp. 1, 3-5.

Jaap, J. P. and E. K. Davis (1988), "The Scheduling Techniques of ESP2," *Proceedings of the Second Annual Workshop on Space Operations Automation and Robotics*, Dayton, Ohio, pp. 1-6.

✓ Jaap, J. P. and E. K. Davis (1989), "Experiment Scheduling Program User's Manual," *NASA George C. Marshall Space Flight Center*, Huntsville, Alabama.

Johnston, M. D. and H.-M. Adorf (1992), "Scheduling with Neural Networks - The Case of The Hubble Space Telescope," *Computers and Operations Research*, Volume 19, No. 3/4, pp. 209-240

Kacprzyk, J. and M. Fedrizzi (1988), "A 'Soft' Measure of Consensus in the Setting of Partial (Fuzzy) Preferences," *European Journal of Operational Research*, Volume 34, pp. 316-325.

Kelley, J. E., Jr. (1963), "The Critical Path Method: Resources Planning and Scheduling," Ch. 21 in *Industrial Scheduling*, J.F. Muth and G.L. Thompson (Eds.), Prentice-Hall, Englewood Cliffs, New Jersey, pp. 347-365.

Machuca, L. F. and R. P. Sadowski (1981), "A Resource Constrained Scheduling System with Time Windows," *Report Number 124, Purdue Laboratory for Applied Industrial Control*, Purdue University, Indiana.

Maxwell, T. G. (1987), "Investigation of Experiment Selection Order Determination for Space Mission Scheduling," Unpublished Master's thesis, University of Alabama, Huntsville.

✓ Miller, G. and M. Johnston (1991), "Long Range Science Scheduling for the Hubble Space Telescope," *Proceedings of the 1991 Goddard Conference on Space Applications of Artificial Intelligence*, Maryland, pp. 71-82. *7/1/91*

Miller, G., M. Johnston, S. Vick, J. Sponsler and Lindenmayer K. (1988), "Knowledge Based Tools for Hubble Space Telescope Planning and Scheduling: Constraints and Strategies," *Proceedings of the 1988 Goddard Conference on Space Applications of Artificial Intelligence*, Greenbelt, Maryland, pp. 91-106. *87A21806*

- ✓ Mission Planning Division (1993), "ESP Requirements: Software Requirements Specification for the Experiment Scheduling Program," *Document Number ESS-SRS-02, Mission Operations Laboratory*, George C. Marshall Space Flight Center, NASA, Huntsville, Alabama.
- Pritsker, A. A. B., L. J. Watter, and Wolfe, P. M. (1969), "Multiproject Scheduling with Limited Resources: A Zero-One Programming Approach," *Management Science*, Volume 16, No. 1.
- Sadowski, R. P., and R. E. Jacobson (1978), "Scheduling Algorithms for the Unbalanced Production Line: An Analysis and Comparison," *AIIE Transactions*, Volume 10.
- Sheskin, T. J. (1988), "Scheduling Experiments on the Space Station," *Computers and Industrial Engineering*, Volume 14, No. 3, pp. 315-323.
- Smith, S. F. and D. K. Pathak (1992), "Balancing Antagonistic Time and Resource Utilization Constraints in Over-Subscribed Scheduling Problems," *Proceedings of the 8th IEEE Conference on Artificial Intelligence for Applications*, Monterey, California, pp. 113-119.
- ✓ Stacy, K. L. and J. P. Jaap (1988), "Space Station Payload Operations Scheduling with ESP2," *Proceedings of the Second Annual Workshop on Space Operations Automation and Robotics*, Dayton, Ohio, pp. 7-13. *8/10/1987*
- ✓ Thalman, N. E., T. P. Sparn, L. Jaffres, D. Gablehouse, D. Judd and Russell C. (1991), "AI Techniques for a Space Application Scheduling Problem," *Proceedings of the 1991 Goddard Conference on Space Applications of Artificial Intelligence*, Maryland, pp. 83-93. *9/14/2006*
- Ullman, J. D. (1976), "Complexity of Sequencing Problems," In *Computer and Job-Shop Scheduling Theory* (Coffmann, E.G., Ed.), John Wiley, New York, pp. 139-164.
- Yamamoto, M. (1977), "An Approximate Solution of Machine Scheduling Problems by Decomposition Method," *International Journal of Production Research*, Volume 15, No. 6, pp. 599-608.